

System Test Plan for Source Code Analysis Tool

Oğuzhan Yangöz, Ashwin Kishore, Yitao Wei

George Washington University

CSCI 6231 Software Engineering

May 4, 2022

1. Introduction

This document provides guidance for the software engineers working on the Source Code Analysis tool for testing the software developed in accordance with the System Requirements Document (SRD) and System Design Document (SDD).

1.1. Objectives

The objectives of the System Test Plan is to ensure ;

- the quality of the software meets the standards set by the stakeholders,
- the behavior of the software is as outlined in the requirements from the discrete units to the whole system,
- to detect and fix many potential software errors before deployment,
- the increased reusability of the software for the future,

2. Document References

SRD: System Requirements Document

SDD: System Design Document

3. System Summary

The SCA tool consists of five subsystems ; User Interface, Query, Report, Data Entry and Functional Analysis subsystems. The SCA tool aims to utilize all these subsystems to assist developers with capturing repository details such as classes, variables, methods, data types and detailed descriptions of the project's components through a simple, yet user-friendly interface. The data of the SCA tool is stored in MongoDB for its compatibility and feasibility for documentation and higher scalability.

4. Test Plan Environment

TBD

5. Major Test Specifications

Major test specifications can be viewed in the next pages below.

George Washington University - SCA Tool

Test Case Specification - 1

Test Case Name: User Log-in Functionality

Description: This test checks the log-in and session start functionality of the SCA tool (SDD 5.2.1 Session & 5.2.2 User). The test will include the following;

- whether log-in window is displayed to the user upon running the application,
- whether the entered credentials are sent to the database for verification, and grants access to the user, redirecting them to the main user interface (SDD Figure 11) if there is a match for both username and password.
- whether the program displays a pop-up window saying *"Invalid username or password. Please check your credentials and try again."* when any of the entered credentials do not match. Note that the error message must not specify whether email or password was entered incorrectly.
- whether the program displays a pop-up window saying *"No credentials entered.Username and password areas cannot be left blank"* if the user attempts to log-in with blank username and password fields without entering any credentials.
- whether the text field *"forgot your password?"* text redirects users to a new pop-up window asking users to enter their username or email, displaying the message *"If the username/email matches our records, we will send you a password-reset email shortly."* upon pressing the "Reset" button regardless of the verification outcome.
- whether the user session will be initiated and displayed continuously on the *"Session Information"* window upon successful log-in.

Internet connection required.

Prepared by: Oğuzhan Yangöz

Components to be Tested: User & Session Component (SDD 5.2.1 Session & 5.2.2 User)

Testing Procedure

1. Testing will start with running the application and waiting for the log-in window to be displayed.
2. Then, an incorrect combination of username and password from the database will be entered as an input to the corresponding fields in the log-in window. The behavior of the system will be observed if the expected error message will be displayed by the system, and user authentication will fail, preventing the user from accessing the system.
3. Another log-in attempt will be made using a registered email address with an incorrect/unmatching password and the system will be observed as to whether it will display the same error message described above and grant access to the user.
4. "Forgot password?" text will be clicked, checking if the user is redirected to a new window asking their username/email address to start the password reset process. The system will be observed as to whether the message described above will be displayed, regardless of the outcome of the verification check.
5. A valid combination of username and password will be entered, and the system will be observed as to whether the user is granted access to the main user interface, and the user session is initiated and continuously displayed in the session information window of the main user interface (SDD Figure 11).

Expected Results

1. When a user runs the application, they will be able to view the log-in window without any error upon running the SCA tool.
2. When a user enters invalid/incorrect combination of username/email and password, the system will display a pop-up window saying *"Invalid username or password. Please check your credentials and try again."* and prevent the user from accessing the main user interface.
3. When a user leaves the username/email and/or password fields blank, the system will display a pop-up window saying *"No credentials entered. Username and password areas cannot be left blank"* and prevent the user from accessing the main user interface.
4. When a user clicks on "Forgot password?" text in the log-in window, the user will be required to enter their email/username and click the "Reset" button. Then, the system will check if the entered email/username matches the records in the database. Regardless of the outcome, the system will display a pop-up window saying *"If the username/email matches our records, we will send you a password-reset email shortly."* If there is a match, the users will be sent a password reset email to their email address in three(3) minutes, asking them to enter a new password. When they submit their new password, their previous password in the database will be replaced with the new one.
5. When a user enters their credentials correctly, the system will redirect the user to the main user interface, starting their session and updating the timer and session information in the "Session Information" window.

Note: User log-out and session termination will be tested in a separate test.

George Washington University - SCA Tool

Test Case Specification - 2

Test Case Name: User Log-out Functionality

Description: This test checks the log-out and session termination functionality of the SCA tool. The test will include the following;

- whether the user session is terminated upon the user's interaction with the log-out button or window management of the operating system.
- whether the session log details (start, end, duration) are logged into the database for future records
- whether the user's log-in status is captured and it is ensured that the user is already logged in before terminating user session and logging the user out of the tool

Internet connection and user authentication required.

Prepared by: Oğuzhan Yangöz

Components to be Tested: User Interface - User & Session (SDD 5.2.1 Session & 5.2.2 User)

Testing Procedure

1. A successfully logged-in user will attempt to click the log-out button in the main user interface(SDD Figure 11), and the system behavior regarding whether the session is terminated, the user is logged out and the user is redirected to the log-in window will be observed.
2. A successfully logged-in user will attempt to shut down the SCA tool using the window management of the operation system, and the system behavior whether the session is terminated and the user is logged out of the system will be observed.
3. A successfully logged-in user will attempt to force close the SCA tool using the activity monitor, and the system behavior whether the session is terminated and the user is logged out of the system will be observed.

Expected Results

1. When a user attempts to click the log-out button, the system will terminate the session, log the user out of the system, append the user log records in the database with the start,end, duration details of the user session to the existing user logs.
- 2 & 3. When a user shuts down the SCA tool using the window management of the operation or force closes the tool using the activity monitor, the system will still be able to terminate the session, log the user out of the system and append the user log records in the database successfully.

George Washington University - SCA Tool

Test Case Specification - 3

Test Case Name: Interacting with Repositories

Description: This test checks the functionalities that allows users to interact with repositories and make changes on them. The test will include the following;

- whether users can open an existing repository that exists in the database,
- whether autosave function captures all changes made in the repository at all times on condition that a stable internet connection is maintained,
- whether users can close a repository without any data/change loss thanks to autosave functionality

Internet connection and user authentication required.

Prepared by: Oğuzhan Yangöz

Components to be Tested: User Interface - File (SDD 5.1.1)

Testing Procedure

1. A logged in user will access a repository through the <File> menu -< open a repository> item. Then, the user will be prompted to type the repository name to be accessed, where the user will enter a non-existing repository name. The system behavior will be observed whether the repository will be opened or an error message will be displayed. (See expected results for the error message details)
2. Then, a logged in user will enter a valid repository name as an input, and the system behavior will be observed as to whether the repository will be viewed in the file view menu in list view.
3. A logged in user who has a repository open in their file view will add a small piece of random text to a file in the repository, close the file view /repository by clicking the x icon located in the upper left corner of the tab, check if the repository will be closed and the user will be taken back to the empty file view menu on the main user interface.
4. The database will be checked to find out if the changes made were successfully captured and updated in the database.

Expected Results

1. The system will display a pop-up window saying *"Repository not found"*.
2. The system will display the requested repository in the list view successfully without any errors in the file view of the main user interface.
3. The repository will be successfully closed, and the user will be returned to the main user interface with an empty file view.
4. The database will have the most recent version of the file, including the newly added string.

George Washington University - SCA Tool

Test Case Specification - 4

Test Case Name: Interacting with Toolbar

Description: This test checks the functionalities that allows users to interact with repositories and make changes on them. The test will include the following;

- whether users can open a new tab and display two files from the same repository side-by-side with an equal share of the window size in the file view window (Figure 11).
- whether users will be displayed a warning message when they open more than two files side-by-side.
- whether users can undo the changes in the reverse order they were made (First in Last out)
- whether users can highlight the changes they made since they opened the file.
- whether users can add comments to the rightmost part of the file view by selecting a piece of text and clicking the annotation option in the toolbar.

Internet connection and user authentication required.

Prepared by: Oğuzhan Yangöz

Components to be Tested: User Interface - Toolbar (SDD 5.1.2)

Testing Procedure

1. A logged in user will open two files with the side-by-side option in the toolbar from the same repository, and check if the files are viewed correctly (See expected results 1)
2. The user will open another file using the side-by-side view option in the toolbar and check if the system will display any error message.
3. The user will open a file in a repository and start making the following changes;
 - adding the text “this is a test for undoing changes”
 - removing the letters one-by-one starting from the end of the sentence until all the entire sentence is removed.
 - clicking the undo icon in the toolbar and observing if every press brings back the letters one by one in each press starting from the beginning of the sentence until the entire sentence is brought back.
 - continuing to undo the changes until the original version of the document before the changes is retrieved and checking if the undo icon is then grayed out.
4. The user will add a random text of any size to one of the files and click the highlight icon in the toolbar and check whether the background color of the changes made since the last opening of the file will be set to yellow.
5. The user will then delete the change they made and click the highlight icon again and observe the system.
6. The user will select a piece of text and click the annotate option from the toolbar, type a random character, hit the return key and check if the comment cloud is displayed.
7. The user will repeat the first part of the step 6 and add a random set of characters of size 5 as a comment, hit return key, and check if the comment was displayed at the rightmost edge of the file corresponding to the text’s position.
8. The user will repeat step 6 and add a long text (30, 50, 100, 250 characters respectively) and check if the comment cloud box size and text input is dynamically aligned and adjusted depending on the input text size.

9. The user will repeat step 6 and hit the return key without typing anything and observe the system behavior.

Expected Results

1. Two files will be displayed successfully without any error or visual glitch in the file view.. Each file will take up the 50 percent of the file view window, displaying the most recent version of the selected files side-by-side.
2. The system will display a pop-up window saying *"You can only view two files side-by-side"* and stop execution.
3. The system will successfully undo all the changes, bringing back the text first, then removing it again the last step, resulting in a grayed out undo button in the toolbar.
4. The system will highlight the changes made by the user since the last opening of the document.
5. The system will display a pop-up message saying *"No changes detected."*
6. The system will display the cloud-shaped comment box with the character typed/entered by the user aligned with the selected text in the rightmost edge of the file view.
7. The system will display the cloud-shaped comment box with the text input typed/entered by the user aligned with the selected text in the rightmost edge of the file view.
8. The system will dynamically display the cloud-shaped comment box without any visual glitch and misalignment regardless of the input size.
9. The system will automatically cancel/remove the cloud-shaped comment box without any error.

George Washington University - SCA Tool

Test Case Specification - 5

Test Case Name: Running Commands through Command Line Interface

Description: This test checks the command line functionalities that allows users to interact with repositories through the command line. The test will include the following;

- whether a command line interface is ready for accepting user commands
- whether an entered command can be run/executed through the command line interface
- whether correct error messages and error codes are displayed when an error occurs because of user-related and internet-related factors.

Internet connection and user authentication required.

Prepared by: Oğuzhan Yangöz

Components to be Tested: User Interface - Command Line (5.1.3)

Testing Procedure

1. On the main user interface, the user checks if the command line interface is accepting any inputs as a command to run.
2. The user types a random string as a command, and observes if any error message is printed, and what is printed as an error message (if any).
3. The user tries a built-in OS function such as cd and specifies a directory, and checks whether the terminal returns a line from the target directory, ready for a new input.
4. The user tries to open a non-existing filename from the specified directory through the terminal, and checks whether the file is opened or an error message is displayed.
5. The user disconnects their computer from the internet and tries to access the repository through the command line interface, and checks whether an error message is displayed.
6. A user with the assigned role of "developer/engineer" tries to access a manager-only file through the command line interface and checks whether the access is given or an error message is displayed.

Expected Results

1. Command line interface is fully functional and ready for a user command/input.
2. Command line interface will display the error *"001 Invalid command – Command not found"*
3. Command line interface will display the error *"002 Invalid directory" – "Directory not found"*
4. Command line interface will display the error *"003 Invalid filename – File not found"*.
5. Command line interface will display the error *"004 Connection Error – Connection to the database lost"*.
6. Command line interface will display the error *"005 Authorization Error – User not authorized (e.g., manage vs developer)"*

George Washington University - SCA Tool

Test Case Specification - 6

Test Case Name: Filtering result

Description: This test checks filtering functionalities that allow users to filter results by some conditions to get their interested information. The test will include the following;

- whether the results are refiltered by only a single filtering condition.
- whether the results are refiltered by only multiple filtering conditions.
- whether the filtering conditions are corresponding to the factors in current results.
- whether correct error messages and error codes are displayed when the filtering is not successful.

Internet connection and user authentication are required.

Prepared by: Yitao Wei

Components to be Tested: Report subsystem - results filtering (SDD 5.4)

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to get detailed and comprehensive results.
2. The tester should check if the conditions are corresponding to the factors shown in the result. If possible, the tester should create different results to see if the conditions will be changes.
3. The tester should select every single condition to see if filtering is applied in the both tree diagram and flame diagram. The tester should expand all branches or notes to guarantee the filtering operations are applied to all results.
4. The tester should select multiple conditions with different combinations to see if filtering is applied in the both tree diagram and flame diagram. The tester should expand all branches or notes to guarantee the filtering operations are applied to all results.

Expected Results

1. When the tester enters the result interface, the diagrams should be initiated based on the current results.
2. The number of conditions is the same as the number of factors shown in the results. And the name of each condition matches the name of each factor shown in the results.
3. The tree diagram and flame diagram can be reconstructed and redisplayed correctly for both single filtering conditions and multiple conditions after the filtering operations.
4. If the filtering operations are unsuccessful, the system will display a pop-up message saying *"Unsuccessful filtering."*

George Washington University - SCA Tool

Test Case Specification - 7

Test Case Name: Information and structure in the tree diagram and flame diagram.

Description: This test checks diagram functionalities that visualize the results in detailed, clear, and well-structured diagrams. The test will include the following;

- whether all results are displayed in the both tree diagram and flame diagram.
- whether all results are structured properly in the both tree diagram and flame diagram.
- whether each node of results displayed in the tree diagram can be expanded and collapsed.
- whether correct error messages and error codes are displayed when the results display is not successful.

Internet connection and user authentication are required.

Prepared by: Yitao Wei

Components to be Tested: Report subsystem - drawing diagram(SDD 5.4)

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to get detailed and comprehensive results.
2. The tester should compare the information in the both tree diagram and flame diagram with the information in the test environment which the tester has already known before the test.
3. The tester should check the name of each node and try to collapse and expand each node to see if the tree diagram responds correctly based on the tester's operations.
4. The test should check if the color of the flame diagram corresponds to the CPU utilization.
5. The test should check if plateaus are corresponding to the number of function calls.

Expected Results

1. When the tester enters the result interface, the diagrams should be initiated based on the current results.
2. All results are displayed and structured in the both tree diagram and flame diagram.
3. The tree diagram responds correctly with expand and collapse operations.
4. The colors of the flame diagram are corresponding to the CPU utilization. The darker the color, the higher the CPU utilization.
5. Plateaus are corresponding to the number of function calls. The wider the plateaus, the more times this function is called.
6. If the filtering operations are unsuccessful, the system will display a pop-up message saying *"Incorrect results."*

George Washington University - SCA Tool

Test Case Specification - 8

Test Case Name: Report export.

Description: This test checks export functionalities that are used to export the report. The test will include the following;

- whether the results can be exported properly with or without filtering conditions.
- whether the results can be exported by each format shown in the format list.
- whether the contents in the report are correct.
- whether correct error messages and error codes are displayed when report export is not successful.

Internet connection and user authentication are required.

Prepared by: Yitao Wei

Components to be Tested: Report subsystem - report export (SDD 5.4)

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to get detailed and comprehensive results.
2. The tester should try to export the report without choosing any conditions.
3. The tester should try to export the report with different combinations of filtering conditions.
4. The tester should try to export the report with each format shown in the format list.
5. The tester should open each report to check if the contents are corresponding to the test results.

Expected Results

1. Reports are able to be generated with or without filtering conditions.
2. The contents of each report match the test results.
3. Reports are able to be generated in different formats.
4. If the export operations are unsuccessful, the system will display a pop-up message saying *"Failed to export report."*

George Washington University - SCA Tool

Test Case Specification - 9

Test Case Name: Command check

Description: This test checks command check functionalities that are used to check the command format and object's existence. The test will include the following;

- whether the command meets a certain format.
- whether the object's existence is checked before command implementation.
- whether correct error messages and error codes are displayed when the command is not correct and the object exists.

Internet connection and user authentication are required.

Prepared by: Yitao Wei

Components to be Tested: DataEntry subsystem - command check(SDD 5.5)

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to create a test environment.
2. The tester should enter different types of the wrong commands to test the format check function.
3. The tester should enter the command to add an existed object to check if there are any warnings.
4. The tester should enter the command to delete a nonexistent object to check if there are any warnings.

Expected Results

1. If the command format is not correct, the system will display a pop-up message saying *"Incorrect format."*
2. if the object exists when the user wants to add it, the system will display a pop-up message saying *"Object existence."*
3. if the object doesn't exist when the user wants to delete it, the system will display a pop-up message saying *"Nonexistent object."*

George Washington University - SCA Tool

Test Case Specification - 10

Test Case Name: Changing object

Description: This test checks object change functionalities that are used to change the value or attributes for all elements in the system. The test will include the following;

- whether the add operations are successful or not.
- whether the delete operations are successful or not.
- whether the modifying operations are successful or not.
- whether correct error messages and error codes are displayed when operations are unsuccessful.
- whether correct messages for successful operations are displayed when operations are successful.

Internet connection and user authentication are required.

Prepared by: Yitao Wei

Components to be Tested: DataEntry subsystem - the modifying object(SDD 5.5)

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to create a test environment.
2. The tester should try to add, delete and change objects that have already existed or not. The tester can use the query interface to see if the operations are successful or not.
3. The tester should try step-by-step commands("What classes does subsystem <Report> have?", "What attributes does class <GetData> have") and direct commands(What attributes does class<GetData> have in subsystem <Report>)

Expected Results

1. If an object which has functions, attributes, or methods is added, those affiliated elements should be added as well.
2. If an object which has functions, attributes, or methods is deleted, those affiliated elements should be deleted as well.
3. If the changing operations are successful, the corresponding object should be changed according to the operation.
4. For the step-by-step commands, the target results are obtained by certain order.
5. For the direct commands, the target results are obtained directly.
6. If the operations are successful, the system will display a pop-up message saying "*Operate successfully.*"
7. If the operations aren't successful, the system will display a pop-up message saying "*Operate unsuccessfully.*"

George Washington University - SCA Tool

Test Case Specification - 11

Test Case Name: Existence Checking

Description: This test checks if the entered data exists in the data store. This allows to verify if the data entered matches with the existing database.

The test will include the following;

- whether the input data exists in the database.
- whether the add operation can be performed for the input which is matching the database.
- whether the delete and modify operations can be performed for the input.

Internet connection and user authentication are required.

Components to be Tested: Data Entry subsystem - Existence (SDD 5.6)

Prepared by: Ashwin Kishore

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to get detailed and comprehensive results.
2. The tester should check if the given input is a valid one by checking if the system can find the input in the data store .
3. The tester should check if the add operation can be performed by the system checking the object existence.
4. The tester should check if the delete and modify operations can be performed by the system checking the object existence.

Expected Results

1. When the input matches with the data in the data store, For the add operation, object existence will be checked first. If not, add operation is performed, otherwise is unsuccessful and pops up the warning dialog "Input found in database" or "Unsuccessful operation".
2. For the delete and modify operations, if the object exists, the operation is successful otherwise is unsuccessful and a warning dialog "Unsuccessful operation"
3. If the input is found in the datastore, the system will display a pop-up message saying "Input found in database"
4. If the operations are successful, the system will display a pop-up message saying "Successfully added"
5. If the operations are successful, the system will display a pop-up message saying "Successfully modified"

George Washington University - SCA Tool

Test Case Specification - 12

Test Case Name: Storing data

Description: This test checks storing functionalities that allow users to store the data by some conditions for the system to create an entry for each attribute of a class and records the data type of each attribute in a class. The test will include the following;

- whether the input results is a valid input by the user.
- whether it creates appropriate entries for each attribute of a class.
- whether it records the data type of each attribute in a class.

Internet connection and user authentication are required.

Components to be Tested: Functional Analysis subsystem - Data store (SDD 5.6)

Prepared by: Ashwin Kishore

Testing Procedure

1. In order to create a test environment, The tester should run the SCA tool for a test program to get detailed and comprehensive results.
2. The tester should check the given input is a valid one which has appropriate attributes and data types by using the error message.
3. The tester should check if appropriate entries for each attribute of a class are being created by the function when it is called by using the error message.
4. The tester should check if it records the appropriate data type of each attribute in a class being created by the function when it is called by using the error message.

Expected Results

1. When the input is valid, it creates an entry and records all the datatypes and pops a message "entry created".
2. If there is no entry the error message pops up saying there is "no data available"
3. If the entries for each attribute of a class are being created by the function it pops "Entry created message" else "Entry could not be created"
4. If the operations are successful, the system will display a pop-up message saying "Data Successfully Recorded"

6. Testing Schedule

TBD

7. Summary of Results - System Test Report

TBD - The results of the test are required to generate a System Test Report.